

1 **CLAIMS**

2 1. A method comprising:

3 sorting, using multiple depth buffers, depth data associated with multiple
4 transparent pixels that overlie one another to identify an individual pixel that lies
5 closest to an associated opaque pixel;

6 computing a transparency effect of the identified pixel relative to the
7 associated opaque pixel; and

8 after said computing, identifying a next closest transparent pixel relative to
9 the opaque pixel and computing, for the next closest pixel, a transparency effect
10 relative to the transparency effect that was just computed.

11
12 2. The method of claim 1, wherein said multiple depth buffers comprise z
13 buffers.

14
15 3. The method of claim 1, wherein said multiple depth buffers comprise w
16 buffers.

17
18 4. The method of claim 1, wherein said multiple depth buffers comprise 1/w
19 buffers.

20
21 5. The method of claim 1, wherein said multiple depth buffers comprise 1/z
22 buffers.

1 6. The method of claim 1, wherein said act of sorting comprises:
2 identifying one of the multiple buffers as a destination buffer that is both
3 readable and writable;
4 identifying another of the multiple buffers as a source buffer that is only
5 readable; and

6 flipping which of the multiple buffers is considered as the destination
7 buffer and the source buffer during said acts of sorting, computing and identifying.

8
9 7. The method of claim 1 further comprising repeating said act of identifying
10 for any additional overlying transparent pixels.

11
12 8. A computing system configured to implement the method of claim 1.

13
14 9. An apparatus comprising:

15 means for sorting, using multiple depth buffers, depth data associated with
16 multiple transparent pixels that overlie one another to identify an individual pixel
17 that lies closest to an associated opaque pixel;

18 means for computing a transparency effect of the identified pixel relative to
19 the associated opaque pixel; and

20 means for identifying a next closest transparent pixel relative to the opaque
21 pixel and computing, for the next closest pixel, a transparency effect relative to the
22 transparency effect that was just computed.

1 **10.** The apparatus of claim 9, wherein said means for sorting and means for
2 identifying comprises hardware comparison logic.

3
4 **11.** A method comprising:

5 (a) rendering at least one opaque pixel that lies along a ray;
6 (b) identifying a transparent pixel that lies along the ray, the identified
7 transparent pixel being the closest transparent pixel to the opaque pixel;
8 (c) computing transparency effects of the identified transparent pixel
9 relative to the opaque pixel;
10 (d) if additional transparent pixels lie along the ray, identifying a next
11 closest transparent pixel relative to the opaque pixel and computing transparency
12 effects of the next closest pixel relative to the computed transparency effects of a
13 last computed transparent pixel; and
14 (e) repeating act (d) until transparency effects of all of the transparent
15 pixels along the ray have been computed in a back-to-front manner.

16
17 **12.** The method of claim 11, wherein acts (a)-(e) are performed utilizing two
18 physical depth buffers for sorting depth data associated with the transparent pixels.

19
20 **13.** The method of claim 11, wherein acts (a)-(e) are performed utilizing two
21 physical depth buffers for sorting depth data associated with the transparent pixels,
22 and wherein the two depth buffers are configured to be flipped.

1 **14.** The method of claim 11, wherein acts (a)-(e) are performed utilizing two
2 physical z buffers for sorting depth data associated with the transparent pixels.
3

4 **15.** The method of claim 11, wherein acts (a)-(e) are performed utilizing two
5 physical w buffers for sorting depth data associated with the transparent pixels.
6

7 **16.** The method of claim 11, wherein acts (a)-(e) are performed utilizing two
8 physical 1/w buffers for sorting depth data associated with the transparent pixels.
9

10 **17.** The method of claim 11, wherein acts (a)-(e) are performed utilizing two
11 physical 1/z buffers for sorting depth data associated with the transparent pixels.
12

13 **18.** The method of claim 11, wherein acts (a)-(e) are performed utilizing two
14 physical z buffers for sorting depth data associated with the transparent pixels, and
15 wherein the two z buffers are configured to be flipped.
16

17 **19.** The method of claim 11, wherein acts (a)-(e) are performed utilizing two
18 physical w buffers for sorting depth data associated with the transparent pixels,
19 and wherein the two w buffers are configured to be flipped.
20

21 **20.** The method of claim 11, wherein acts (a)-(e) are performed utilizing two
22 physical 1/w buffers for sorting depth data associated with the transparent pixels,
23 and wherein the two 1/w buffers are configured to be flipped.
24

1 **21.** The method of claim 11, wherein acts (a)-(e) are performed utilizing two
2 physical 1/z buffers for sorting depth data associated with the transparent pixels,
3 and wherein the two 1/z buffers are configured to be flipped.

4

5 **22.** The method of claim 11, wherein acts (a)-(e) are performed utilizing two
6 physical depth buffers for sorting depth data associated with the transparent pixels,
7 and wherein performing acts (a)-(e) comprise:

8 designating one of the depth buffers as readable and writable;
9 designating the other of the depth buffers as readable only; and
10 flipping the designations of the depth buffers.

11

12 **23.** The method of claim 11, wherein acts (a)-(e) are performed utilizing two
13 physical z buffers for sorting depth data associated with the transparent pixels, and
14 wherein performing acts (a)-(e) comprise:

15 designating one of the z buffers as readable and writable;
16 designating the other of the z buffers as readable only; and
17 flipping the designations of the z buffers.

18

19 **24.** The method of claim 11, wherein acts (a)-(e) are performed utilizing two
20 physical w buffers for sorting depth data associated with the transparent pixels,
21 and wherein performing acts (a)-(e) comprise:

22 designating one of the w buffers as readable and writable;
23 designating the other of the w buffers as readable only; and
24 flipping the designations of the w buffers.

1 **25.** The method of claim 11, wherein acts (a)-(e) are performed utilizing two
2 physical 1/w buffers for sorting depth data associated with the transparent pixels,
3 and wherein performing acts (a)-(e) comprise:

4 designating one of the 1/w buffers as readable and writable;
5 designating the other of the 1/w buffers as readable only; and
6 flipping the designations of the 1/w buffers.

7
8 **26.** The method of claim 11, wherein acts (a)-(e) are performed utilizing two
9 physical 1/z buffers for sorting depth data associated with the transparent pixels,
10 and wherein performing acts (a)-(e) comprise:

11 designating one of the 1/z buffers as readable and writable;
12 designating the other of the 1/z buffers as readable only; and
13 flipping the designations of the 1/z buffers.

14
15 **27.** A computing system configured to implement the method of claim 11.

1 **28.** A system comprising:

2 means for rendering at least one opaque pixel that lies along a ray;

3 means for identifying a transparent pixel that lies along the ray, the

4 identified transparent pixel being the closest transparent pixel to the opaque pixel;

5 means for computing transparency effects of the identified transparent pixel

6 relative to the opaque pixel; and

7 means for identifying, in a back-to-front manner, additional transparent

8 pixels and successively computing transparency effects for the additional

9 transparent pixels.

10

11 **29.** The system of claim 28, wherein said means for rendering comprises a

12 graphics subsystem.

13

14 **30.** The system of claim 28, wherein said means for identifying comprises a

15 pair of physical depth buffers.

16

17 **31.** The system of claim 28, wherein said means for identifying comprises a

18 pair of physical depth buffers that can be logically flipped.

- 1 32. A system comprising:
- 2 a transparent depth sorting component comprising:
- 3 at least two physical depth buffers;
- 4 a writeback counter to count writebacks that occur to at least one of
- 5 the two physical depth buffers; and
- 6 comparison logic that is configured to effect:
- 7 sorting, using said at least two physical buffers, of depth data
- 8 associated with multiple transparent pixels that overlie one another
- 9 to identify an individual pixel that lies closest to an associated
- 10 opaque pixel;
- 11 computing a transparency effect of the identified pixel
- 12 relative to the associated opaque pixel;
- 13 after said computing, identifying a next closest transparent
- 14 pixel relative to the opaque pixel; and
- 15 computing, for the next closest pixel, a transparency effect
- 16 relative to the transparency effect that was computed for the said
- 17 closest individual pixel and the associated opaque pixel.

1 **33.** The system of claim 32, wherein:

2 one of said at least two physical depth buffers is capable of being
3 designated as readable and writable;

4 another of said at least two physical depth buffers is capable of being
5 designated as readable only; and

6 designations of said at least two physical depth buffers are capable of being
7 flipped.

8

9 **34.** The system of claim 32, wherein said at least two physical depth buffers
10 comprise z buffers.

11

12 **35.** The system of claim 32, wherein said at least two physical depth buffers
13 comprise w buffers.

14

15 **36.** The system of claim 32, wherein said at least two physical depth buffers
16 comprise 1/w buffers.

17

18 **37.** The system of claim 32, wherein said at least two physical depth buffers
19 comprise 1/z buffers.

20

21 **38.** The system of claim 32, wherein said transparent depth sorting component
22 is configured to terminate transparent depth sorting when the writeback counter
23 indicates that no writebacks have occurred.

1 **39.** A graphics subsystem embodying the transparent depth sorting component
2 of claim 32.

3

4 **40.** A computer system embodying the graphics subsystem of claim 39.

5

6 **41.** A method comprising:

7 mapping a first of two depth buffers as a destination buffer that is readable
8 and writable, a second of the two depth buffers being designated as a source buffer
9 that is only readable;

10 rendering one or more opaque objects having associated opaque pixels;

11 writing a depth value associated with an opaque pixel to the first buffer;

12 mapping the second of the depth buffers as the destination buffer, the first
13 of the depth buffers being designated as the source buffer;

14 initializing the destination buffer to a predetermined value;

15 effecting a comparison of a new pixel depth value with values in the source
16 and destination buffers and writing the new pixel depth value to the destination
17 buffer if the new pixel depth value is (a) greater than the value currently in the
18 destination buffer and (b) less than the value in the source buffer, effective to write
19 a new pixel depth value that is associated with a pixel that is closest to a pixel
20 whose depth value is contained in the source buffer;

21 rendering one or more transparent objects having associated transparent
22 pixels;

23 determining if transparency effects for all transparent pixels in all
24 transparent objects have been accounted for and if so, terminating processing and,
25 if not:

1 mapping the first of the depth buffers as the destination buffer, the
2 second of the buffers being designated as the source buffer;

3 effecting a comparison of the new pixel depth value with values in
4 the source and destination buffers and writing to a frame buffer and the
5 destination buffer if the new pixel depth value is equal to the value in the
6 source buffer and the new pixel depth value is less than the value in the
7 destination buffer;

8 rendering one or more transparent objects; and

9 returning to said act of mapping the second of the depth buffers until
10 transparency effects of all transparent pixels in all the transparent objects
11 have been accounted for.

12
13 42. The method of claim 41, wherein said predetermined value comprises a
14 depth buffer's smallest value.

15
16 43. The method of claim 41, wherein said act of determining is performed by
17 maintaining a depth buffer writeback counter that keeps track of depth buffer
18 writebacks.

19
20 44. The method of claim 41, wherein the depth buffers comprise z buffers.

21
22 45. The method of claim 41, wherein the depth buffers comprise w buffers.

23
24 46. The method of claim 41, wherein the depth buffers comprise $1/w$ buffers.
25

- 1 47. The method of claim 41, wherein the depth buffers comprise 1/z buffers.
- 2
- 3 48. A computing system configured to implement the method of claim 41.
- 4
- 5 49. A system comprising:
- 6 a processor;
- 7 at least two depth buffers;
- 8 a frame buffer; and
- 9 a graphics subsystem operably connected with the processor and configured
- 10 to, under the influence of the processor:
- 11 map a first of the depth buffers as a destination buffer that is
- 12 readable and writable, a second of the depth buffers being designated as a
- 13 source buffer that is only readable;
- 14 render one or more opaque objects having associated opaque pixels;
- 15 write a depth value associated with an opaque pixel to the first
- 16 buffer;
- 17 map the second of the depth buffers as the destination buffer, the
- 18 first of the depth buffers being designated as the source buffer;
- 19 initialize the destination buffer to a predetermined value;
- 20 effect a comparison of a new pixel depth value with values in the
- 21 source and destination buffers and write the new pixel depth value to the
- 22 destination buffer if the new pixel depth value is (a) greater than the value
- 23 currently in the destination buffer and (b) less than the value in the source
- 24 buffer, effective to write a new pixel depth value that is associated with a
- 25 pixel that is closest to a pixel whose depth value is contained in the source

1 buffer;

2 render one or more transparent objects having associated transparent
3 pixels;

4 determine if transparency effects for all transparent pixels in all the
5 transparent objects have been accounted for and if so, terminate processing
6 and, if not:

7 map the first of the depth buffers as the destination buffer, the
8 second of the buffers being designated as the source buffer;

9 effect a comparison of the new pixel depth value with values
10 in the source and destination buffers and write to the frame buffer
11 and the destination buffer if the new pixel depth value is equal to the
12 value in the source buffer and the new pixel depth value is less than
13 the value in the destination buffer;

14 render one or more transparent objects; and

15 return to said mapping the second of the depth buffers until
16 transparency effects of all transparent pixels in all the transparent
17 objects have been accounted for.

19 **50.** The system of claim 49, wherein said predetermined value comprises a
20 depth buffer's smallest value.

22 **51.** The system of claim 49 further comprising a depth buffer writeback
23 counter that keeps track of depth buffer writebacks.

25 **52.** The system of claim 49, wherein the depth buffers comprise z buffers.